

Declaring Variables with a Dim Statement

To make the same computer programs work for lots of different data, the program must be written with variables. When the program is run, a user enters specific data.

Question: Where does this data go?

Question: How does the computer know what the variable is for?

Answer: The **variables** that will be used in a program must be **declared**. **Declaring a variable** does two main things. It sets aside a **location in computer memory** that is identified by that variable name. The variable declaration also tells the computer what **type of data** the variable will hold so it reserves the correct amount of space in memory. Different kinds of data use different amounts of memory. The less memory used, the faster the program executes. The data type determines what operations can be done with the data.

Data Types

Data types are what they sound like: types of data. Think of the different types of data users could enter as they run a program. Visual Basic defines eight different types of data that a user could enter or a program could produce.

Why do you need to know about data types? Defining data types for variables gives you a way to control what a user can enter into a textbox. Imagine, for example that you want a user to enter a dollar amount in a textbox labeled “Cost of bike:” You do not want the program to accept other types of data such as text.

First you create a variable to store the data the user enters into this textbox. Then, you can declare the data type of the variable as currency and you can write some code that displays an error message if the user enters the wrong data type in the textbox.

The eight different data types include:

- **Integer** – whole number (no decimals) from -32768 to 32767
- **Long** – integers (no decimals) from -2147483648 to 2147483647
Since the Long data type uses more memory, only use it when Integer is not big enough. Do **NOT** use commas.
- **Single** – decimal numbers, including scientific notation (exponential notation)
- **Double** – decimal numbers with more decimal places
- **Currency** – used for money in dollars (cents as a decimal)
Single can be used but may result in rounding errors. Currency is accurate to one hundredth of a cent.
- **String** – text, including letters, numbers, symbols
- **Variant**
- **User defined**

Declaring Variables with a Dim Statement

Declaring Variables

Declaring a variable means writing a statement that associates a data type with a variable name. This statement announces the existence of the variable and allocates storage for it.

Once the association is made, any value of the variable will be of the declared data type. To declare a variable in Visual Basic, you use a **Dim** statement, as shown in these examples.

```
Dim Address As String
```

```
Dim Length As Single
```

```
Dim Age As Integer
```

```
Dim Principal As Currency
```

```
Dim PerimeterRectangle As Single, Length As Single, Width As Single
```

Textboxes Hold Strings That Can Be Converted into Numeric Data

Textboxes on a form are automatically declared as the `String` data type. Any time this data needs to be used as a number, a new variable will need to be declared in the code with the proper data type and the string from the textbox must be converted into that type using a **Val** statement. After the `Val` statement the variable with the numeric data type can be used in calculations or comparisons.

```
Dim Height As Single, Base1 As Single, Base2 As Single
Height = Val(txtHeight)
Base1 = Val(txtBase1)
Base2 = Val(txtBase2)
```

If the program is going to calculate the area of the trapezoid, you will need a variable to hold the answer. This does not come from a textbox, because the user is not providing it. The program is calculating it. It needs a variable to be declared in a **Dim** statement.

```
Dim AreaTrapezoid As Single
```

Question: Why was the `Single` data type used?

Answer: Length of geometric figures can have decimals.

After the **Val** statement the variable with the numeric data type can be used in calculations or comparisons. The area of the trapezoid can be calculated with an **Assignment Statement**.

```
AreaTrapezoid = (Height * (Base1 + Base2)) / 2
```